

| | |
|--|--|
| Material permitido: Solo calculadora no programable | Aviso 1: Todas las respuestas deben estar debidamente razonadas. |
| Tiempo: 120 minutos | Aviso 2: Escriba con buena letra y evite los tachones. |
| N1 | Aviso 3: Solución del examen y fecha de revisión en https://www.dia.uned.es/71902048/ |

1. Conteste razonadamente a las siguientes preguntas:

- (1 p) Enumerar la *secuencia de eventos* que se producen cuando un programa de usuario invoca a una *llamada al sistema*.
- (1 p) Enumerar y describir brevemente las *regiones* que se diferencian en el espacio de direcciones de *memoria lógica o virtual* de un proceso.
- (1 p) Enumerar y explicar brevemente las cuatro condiciones que se deben cumplir para que exista interbloqueo.
- (1 p) ¿Qué es el *buffering*? ¿Qué problemas resuelve?
- (1 p) Explicar cómo afecta a la eficiencia de un sistema informático el tamaño de bloque de disco que se utilice en un determinado sistema de archivos.

2. (1 p) El baño de caballeros de un centro comercial posee una capacidad para seis caballeros. Cuando el servicio está completo los caballeros que desean pasar deben esperar fuera haciendo cola al lado de la puerta. Además si el operario de limpieza está limpiando el baño no puede pasar ningún caballero. Por otra parte, el operario solo pasa a limpiar el baño si éste está vacío. Escribir el pseudocódigo de un programa que usando **semáforos binarios** coordine la actividad de los caballeros y del operario de limpieza.

Nota 1: Antes de escribir el pseudocódigo se debe explicar adecuadamente el significado de cada una de las variables globales y semáforos binarios que se van a utilizar en el mismo.

Nota 2: En la resolución de este ejercicio se deben utilizar las operaciones para semáforos binarios definidas en el libro base de la asignatura.

3. (2 p) En un sistema x86-64 bits la unidad direccionable es un byte y se suelen usar 48 bits para las direcciones lógicas. Supuesto que la memoria principal tiene un tamaño de 8 GiB y un tamaño de página de 1 KiB:

- (1 p) Determinar el tamaño en bits de cada uno de los campos en que se descompone una dirección física y una dirección lógica.
- (1 p) Supuesto que una entrada de una tabla de páginas tiene un tamaño de 6 bytes, determinar el tamaño que tendría una tabla de páginas completa.

| | |
|--|--|
| Material permitido: Solo calculadora no programable | Aviso 1: Todas las respuestas deben estar debidamente razonadas. |
| Tiempo: 120 minutos | Aviso 2: Escriba con buena letra y evite los tachones. |
| N1 | Aviso 3: Solución del examen y fecha de revisión en https://www.dia.uned.es/71902048/ |

4. (2 p) Un sistema operativo tiene una configuración del tipo *un hilo de núcleo por cada hilo de usuario*. El planificador del sistema operativo utiliza un algoritmo de *turno rotatorio* con un cuanto de 100 ut. El cuanto de 100 ut que se asigna a una tarea se distribuye de forma circular entre los hilos de dicha tarea en rodajas de 10 ut. El tiempo promedio de cambio de tarea (proceso) es de 20 ut y el tiempo promedio de cambio de hilo de una misma tarea es despreciable. Si un hilo tiene que realizar una espera de E/S es porque en la última instrucción de su ráfaga de CPU ha invocado a una llamada al sistema de E/S. En un cierto instante de tiempo se dispone del siguiente conjunto de tareas para su planificación:

- Tarea A, consta de 4 hilos:
 - Hilo A1: requiere de una ráfaga de CPU de 30 ut, una espera de E/S de 110 ut y una ráfaga de CPU de 40 ut.
 - Hilo A2: requiere de una ráfaga de CPU de 50 ut.
 - Hilo A3: requiere de una ráfaga de CPU de 30 ut.
 - Hilo A4: requiere de una ráfaga de CPU de 20 ut, una espera de E/S de 60 ut y una ráfaga de CPU de 40 ut.
- Tarea B, consta de 2 hilos:
 - Hilo B1: requiere de una ráfaga de CPU de 20 ut, una espera de E/S de 50 ut y una ráfaga de CPU de 60 ut.
 - Hilo B2: requiere de una ráfaga de CPU de 40 ut, una espera de E/S de 110 ut y una ráfaga de CPU de 20 ut.

Dibujar el diagrama de uso de la CPU y del canal de E/S suponiendo que el sistema dispone de una CPU y de un canal de E/S que solo puede soportar una única operación de E/S simultáneamente y que el sistema operativo elige primero para planificación a la tarea A.

NOTA: Si un hilo de una tarea se desbloquea al finalizar la operación de E/S que había solicitado entonces se coloca en la cola de hilos preparados para ejecución de dicha tarea en la posición adecuada para que se cumpla la distribución circular en rodajas de 10 ut del cuanto de ejecución de 100 ut. Por lo tanto, en el caso de la tarea A el orden de ejecución de los hilos, si no se encuentran bloqueados o no han finalizado, va a ser:

HA1 → HA2 → HA3 → HA4 → HA1 → HA2 → HA3 → HA4 → ...

SISTEMAS OPERATIVOS (Cód. 71902048)

Solución Examen Enero 2025

Solución Ejercicio 1

- a) Cuando un programa de usuario invoca a una llamada al sistema se produce una *trampa*, que provoca la conmutación hardware de modo usuario a modo supervisor y transfiere el control al núcleo. El núcleo examina el identificador numérico de la llamada (cada llamada tiene asignado uno) y los parámetros de la llamada para poder localizar y ejecutar a la rutina del núcleo asociada a la llamada al sistema. Cuando dicha rutina finaliza, almacena el resultado en algún registro y provoca la conmutación hardware de modo supervisor a modo usuario para que el proceso de usuario que invocó la llamada continúe su ejecución.
- b) El *espacio de direcciones de memoria lógica o virtual de un proceso*, también a veces denominado *imagen del proceso* o *espacio de direcciones de usuario*, se divide en varias *regiones*, cada una de las cuales delimita un área de direcciones contiguas de memoria lógica. De forma general se suelen distinguir al menos tres regiones: la región de código (también denominada región de texto), la región de datos y la región de pila. Adicionalmente pueden existir regiones de memoria compartida y regiones asociadas a librerías.

Las *regiones de código y de datos* se corresponden con las zonas de código y datos del archivo ejecutable. La región de datos se divide en dos regiones: la región de datos inicializados y la región de datos no inicializados. La *región de datos inicializados* tiene un tamaño fijo. Por su parte la *región de datos no inicializados* tiene un tamaño que puede variar durante la ejecución de un proceso.

La *región de pila* se crea automáticamente y su tamaño es ajustado dinámicamente en tiempo de ejecución por el sistema operativo. La pila está constituida por marcos de pila lógicos. Un marco se añade a la pila cuando se llama a una función y se extrae cuando se vuelve de la misma. El *registro puntero de pila* del procesador almacena, dependiendo de la arquitectura del computador, la próxima entrada libre de la pila o la última utilizada. Análogamente, la máquina indica la dirección de crecimiento de la pila, hacia las direcciones altas o bajas.

Un marco de pila contiene usualmente la siguiente información: los parámetros de la función, sus variables locales y las direcciones almacenadas en diferentes registros especiales de la máquina, como por ejemplo, el contador del programa y el puntero de la pila. Salvar el contenido del contador del programa permite conocer la dirección de retorno donde debe continuar la ejecución una vez que se ha ejecutado la función. Mientras que salvar el contenido del registro de pila permite conocer la ubicación del marco de pila anterior o del siguiente libre.

- c) Para que se produzca un interbloqueo se deben cumplir necesariamente las siguientes cuatro condiciones:
1. *Exclusión mutua*. Cada instancia de un recurso solo puede ser asignada a un proceso como máximo.
 2. *Retención y espera*. Cada proceso retiene los recursos que le han sido asignado mientras espera por la adquisición de los otros recursos que necesita.
 3. *No existencia de expropiación*. Si un proceso posee un recurso, éste no se le puede expropiar.
 4. *Espera circular*. Existe una cadena circular de dos o más procesos, de tal forma que cada proceso de la cadena se encuentra esperando por un recurso retenido por el siguiente proceso de la cadena.

- d) El *buffering* es un técnica implementada por el sistema operativo que consiste en almacenar temporalmente en buffers los datos que se transfieren entre un dispositivo y un proceso, o viceversa. Un *buffer* es un área de memoria principal a la que únicamente tiene acceso el sistema operativo, es decir, pertenece al espacio del núcleo.

El buffering resuelve los problemas que plantea la transferencia directa de datos desde el dispositivo al espacio de usuario de un proceso.

- e) Un factor que afecta a la eficiencia del sistema es el tamaño del bloque de disco que se va a utilizar. Si el tamaño de bloque es muy pequeño un archivo puede ocupar muchos bloques, lo que aumentará el número de accesos al disco, con la consiguiente pérdida de tiempo que supone las búsquedas de bloques en un disco magnético. Sin embargo la fragmentación interna dentro de un bloque será muy pequeña con lo que se reduce el desperdicio de espacio. Por otra parte si el tamaño de bloque es muy grande un archivo constará de pocos bloques y se reduce el número de accesos, pero por el contrario la fragmentación interna será grande desperdiciando mucho espacio. En consecuencia, el tamaño de bloque debe ser un compromiso entre el espacio en disco desperdiciado y el número de bloques que se requieren para almacenar un archivo. Diversos estudios han puesto de manifiesto que el tamaño óptimo de bloque debe ser de unos pocos KiB.

Solución Ejercicio 2

La solución que se propone en la Figura 1 para este problema utiliza las siguientes variables globales y semáforos binarios:

- contador. Variable global de tipo entero para llevar la cuenta del número de caballeros.
- S1. Semáforo binario que se utiliza para garantizar la exclusión mutua en el uso de la variable global contador.
- S2. Semáforo binario que se utiliza para sincronizar la actividad del operario de limpieza y los caballeros.
- S3. Semáforo binario que se utiliza para sincronizar la entrada al baño de los caballeros.

```

/* Definición variables y semáforos */
semáforo_binario S1, S2, S3;
int contador=0;

void caballero() /* Proceso caballero */
{
    wait_sem(S3); /* Esperar si el baño está completo */

    wait_sem(S1);
    contador = contador + 1;
    if (contador==1)
    {
        wait_sem(S2); /* Esperar si el operario está en el baño */
        signal_sem(S3); /* Hay puestos en el baño */
    }
    else if (contador < 6) signal_sem(S3); /* Hay puestos en el baño */
    signal_sem(S1);

    usar_baño();

    wait_sem(S1);
    if (contador == 1) signal_sem(S2); /* Avisa al operario para que pase */
    else if {contador == 6} signal_sem(S3); /* Avisa a un caballero para que pase */
    contador = contador - 1;
    signal_sem(S1);
}

void operario_limpieza() /* Proceso operario_limpieza */
{
    wait_sem(S2); /* Esperar si el baño está ocupado */
    limpiar_baño();
    signal_sem(S2); /* Avisa al primer caballero para que pase */
}

void main() /* Inicialización de semáforos y ejecución concurrente */
{
    init_sem(S1,1);
    init_sem(S2,1);
    init_sem(S3,1);
    ejecución_concurrente(caballero,...,caballero,operario_limpieza);
}

```

Figura 1

Solución Ejercicio 3

- a) Cuando se utiliza la técnica de paginación una dirección física se descompone en los campos número de marco de página de f bits y desplazamiento dentro del marco de d bits. Por su parte una dirección lógica se descompone en los campos número de página de p bits y desplazamiento dentro de la página de d bits.

- *Dirección Física:*

Para obtener el tamaño del campo número de marco de página se va a calcular en primer lugar el número de marcos N_{MP} en que se descompone la memoria principal si se considera un tamaño de página $S_P = 4 \text{ KiB}$:

$$N_{MP} = \text{floor}\left(\frac{C_{MP}}{S_P}\right) = \text{floor}\left(\frac{8 \text{ GiB}}{1 \text{ KiB}}\right) = \text{floor}\left(\frac{2^{33}}{2^{10}}\right) = 2^{23} \text{ marcos}$$

Conocido N_{MP} el tamaño de este campo se obtiene resolviendo la siguiente desigualdad:

$$\min_f \{N_{MP} \leq 2^f\}$$

Luego como $N_{MP} = 2^{23}$ entonces $f = 23$ bits.

Para obtener el tamaño del campo desplazamiento, puesto que la unidad direccionable es un byte, hay que resolver la siguiente desigualdad:

$$\min_d \{S_P \leq 2^d\}$$

Luego como $S_P = 2^{10}$ entonces $d = 10$ bits.

| | |
|-----------------------------|--------------------------------------|
| Marco j | Desplazamiento d |
| $f = 23$ bits | $d = 10$ bits |

- *Dirección lógica:*

El tamaño del campo desplazamiento de una dirección lógica es igual al de una dirección física, en este caso, $d = 10$ bits. Por otra parte, en el enunciado se indica que una dirección lógica tiene un tamaño de 48 bits, luego la suma del tamaño p del campo número de página y del tamaño d del campo desplazamiento debe ser igual a 48 bits:

$$p + d = 48$$

Luego $p = 48 - d = 48 - 10 = 38$ bits.

| | |
|------------------------------|--------------------------------------|
| Página i | Desplazamiento d |
| $p = 38$ bits | $d = 10$ bits |

- b) La unidad direccionable es un byte y se suelen usar 48 bits para las direcciones lógicas. En ese caso, el espacio de direcciones lógicas de un proceso puede tener un tamaño máximo de

$$C_X = 2^{48} \text{ B} = 256 \text{ TiB}$$

Supuesto un proceso de este tamaño, si el tamaño de página es $S_P = 1 \text{ KiB} = 2^{10} \text{ B}$, entonces el número de páginas N_P en que se divide el espacio de direcciones lógicas de este proceso es:

$$N_P = \text{ceil}\left(\frac{C_X}{S_P}\right) = \text{ceil}\left(\frac{2^{48}}{2^{10}}\right) = \frac{2^{48}}{2^{10}} = 2^{38} \text{ páginas}$$

La tabla de páginas de este proceso tiene una entrada por cada página, es decir, 2^{38} entradas. Como cada entrada de la tabla de páginas ocupa $S_E = 6$ bytes, entonces la tabla de páginas completa tendría un tamaño de

$$C_{TP} = S_E \times N_P = 6 \times 2^{38} = \mathbf{1536 \text{ GiB}}$$

Solución Ejercicio 4

En la Figura 1 se muestra el diagrama de uso de la CPU y del canal de E/S si se elige primero para planificación a la tarea A.

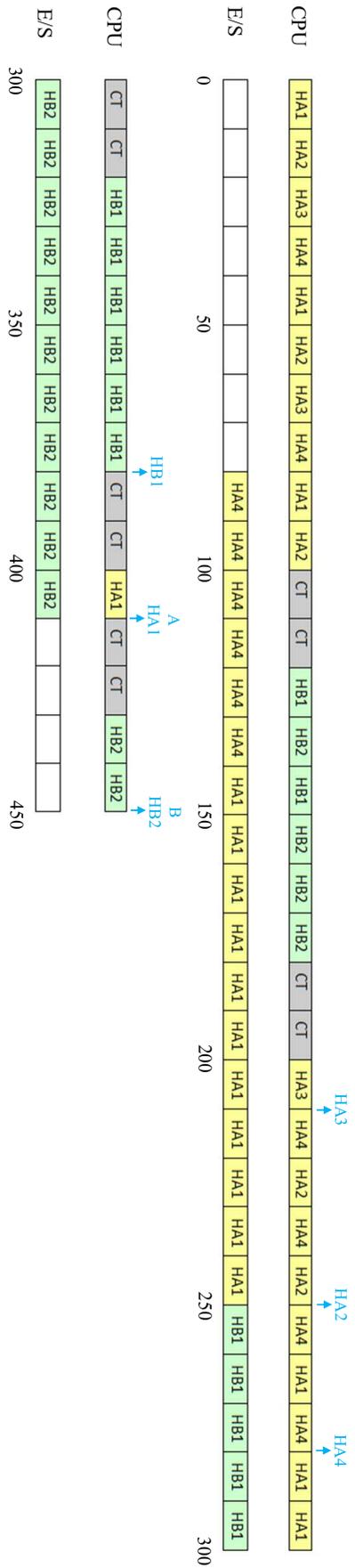


Figura 2